09-29-00

A

EXPRESS MAIL LABEL NO. EM598712491US

Attorney Docket No. 72255/02663

Box Patent Application
Assistant Commissioner for Patents
Washington, DC 20231

NEW APPLICATION TRANSMITTAL

Transmitted herewith for filing is the patent application of

Inventor(s) : Kenneth W. Batcher

For (title) : HARDWARE-BASED ENCRYPTION/DECRYPTION EMPLOYING DUAL PORTED KEY STORAGE

1. Type of Application

This new application is for a(n):

(X) Original (nonprovisional)
( ) Continuation
( ) Continuation-in-part (CIP)
( ) Divisional
( ) Design
( ) Plant

NOTE: If continuation, CIP or divisional, then complete section 2.

CERTIFICATION UNDER 37 C.F.R. 1.10*
(Express Mail label number is mandatory.)
(Express Mail certification is optional.)

I hereby certify that this New Application Transmittal and the documents referred to as attached therein are being deposited with the United States Postal Service on this date September 28, 2000, in an envelope as "Express Mail Post Office to Addressee," mailing Label Number EM598712491US, addressed to the: Assistant Commissioner for Patents, Washington, D.C. 20231.

Valerie A. Milam

Valerie A. Milam

Signature of person mailing paper

WARNING: *Certificate of mailing (first class) or facsimile transmission procedures of 37 C.F.R. 1.8 cannot be used to obtain a date of mailing or transmission for this correspondence.*

****WARNING:*** *Each paper or fee filed by "Express Mail"* ***must*** *have the number of the "Express Mail" mailing label placed thereon prior to mailing. 37 C.F.R. 1.10(b).*

2. Benefit of Prior U.S. Application(s) (35 U.S.C. 119(e), 120, or 121)

NOTE: If the new application being transmitted is a continuation, CIP or divisional, of a parent case, or where the parent case is an International Application which designated the U.S., or the benefit of a prior **provisional** application is claimed, then check the following item and complete section as follows.

() The new application being transmitted claims the benefit of prior U.S. application(s).

2.1 Relate Back

WARNING: If an application claims the benefit of the filing date of an earlier filed application under 35 U.S.C. 120, 121 or 365(c), the 20-year term of that application will be based upon the filing date of the earliest U.S. application that the application makes reference to under 35 U.S.C. 120, 121 or 365(c). (35 U.S.C. 154(a)(2) does not take into account, for the determination of the patent term, any application on which priority is claimed under 35 U.S.C. 119, 365(a) or 365(b).) For a CIP application, applicant should review whether any claim in the patent that will issue is supported by an earlier application and, if not, the applicant should consider canceling the reference to the earlier filed application. The term of a patent is not based on a claim-by-claim approach. See Notice of April 14, 1995, 60 Fed Reg. 20,195, at 20,205.

(complete the following, if applicable)

Amend the specification by inserting, before the first line, the following sentence:

A. 35 U.S.C. 120, 121 and 365(c)

() "This is a
( ) continuation
( ) continuation-in-part
( ) divisional

of copending application(s) serial number filed on ."

( ) International Application________ filed on_______ and which designated the U.S."

Note: *The proper reference to a prior filed PCT application that entered the U.S. national phase is the U S. serial number and the filing date of the PCT application that designated the U.S. Moreover, (1) Where the application being transmitted adds subject matter to the International Application, then the filing can be as a continuation-in-part or (2) if it is desired to do so for other reasons then the filing can be as a continuation.*

( ) "The nonprovisional application designated above, namely application no.______________, filed___________, claims the benefit of U.S. Provisional Application(s) No(s).:

{list application no(s). and filing date(s)}

B. 35 U.S.C. 119(e) (Provisional Application)

( ) "This application claims the benefit of U.S. Provisional Application(s) No(s).:

{list application no(s). and filing date(s)}

2.2 Relate Back—35 U.S.C. 119 Priority Claim for Prior Application

The prior U.S. application(s), including any prior International Application designating the U.S., identified above in item 2.1(A), in turn itself claim(s) foreign priority(ies) as follows:

{list country, application no(s). and filing date(s)}

The certified copy(ies) has (have)

( ) been filed on_____, in prior application serial no.____, which was filed on___.

( ) is (are) attached.

2.3 Maintenance of Copendency of Prior Application

NOTE: The PTO finds it useful if a copy of the petition filed in the prior application extending the term for response is filed with the papers constituting the filing of the continuation application. Notice of November 5, 1985 (1060 O.G. 27).

A. ( ) Extension of time in prior application

*(This item **must** be completed and the papers filed **in the prior application** if the period set in the prior application has run.)*

( ) A petition, fee and response extends the term in the pending **prior** application until Extension of_____.

( ) A **copy** of the petition filed in prior application is attached.

B. ( ) Conditional Petition for Extension of Time in Prior Application

(complete this item, if previous item not applicable)

( ) A conditional petition for extension of time is being filed in the pending **prior** application.

( ) A **copy** of the conditional petition filed in the prior application is attached.

2.4 Further Inventorship Statement Where Benefit of Prior Application(s) Claimed

(complete applicable item A, B and/or C below)

A. ( ) This application discloses and claims only subject matter disclosed in the prior application whose particulars are set out above, and the inventor(s) in this application are

( ) the same.

( ) less than those named in the prior application. It is requested that the following inventor(s) identified for the prior application be deleted:

{ type name(s) of inventor(s) to be deleted }

B. ( ) This application discloses and claims additional disclosure by amendment and a new declaration or oath is being filed. With respect to the prior application, the inventor(s) in this application are

( ) the same.

( ) the following additional inventor(s) have been added:

(type name(s) of inventor(s) to be added)

C. ( ) The inventorship for all the claims in this application are

( ) the same.

( ) not the same. An explanation, including the ownership of the various claims at the time the last claimed invention was made

( ) is submitted.

( ) will be submitted.

2.5 Abandonment of Prior Application *(if applicable)*

( ) Please abandon the prior application at a time while the prior application is pending, or when the petition for extension of time or to revive In that application is granted, and when this application is granted a filing date, so to make this application copending with said prior application.

NOTE: *According to the Notice of May 13, 1983 (103, TMOG 6-7), the filing of a continuation or continuation-in-part application is a proper response with respect to a petition for extension of time or a petition to revive and should include the express abandonment of the prior application conditioned upon the granting of the petition and the granting of a filing date to the continuing application.*

2.6 Petition for Suspension of Prosecution for the Time Necessary to File an Amendment

NOTE: *Where it is possible that the claims on file will give rise to a first action final for this continuation application and for some reason an amendment cannot be filed promptly (e.g, experimental data is being gathered) it may be desirable to file a petition for suspension of prosecution for the time necessary.*

(check the next Item, if applicable)

( ) There is provided herewith a Petition To Suspend Prosecution for the Time Necessary to File An Amendment (New Application Filed Concurrently)

2.7 Small Entity (37 CFR § 1.28(a))

( ) Applicant has established small entity status by the previous submission of a statement in prior application serial no.____ on ____.

( ) A copy of the statement previously filed is included.

WARNING: *See 37 CFR § 1.28(a).*

2.8. Notification in Parent Application of this Filing

( ) A notification of the filing of this
(check one of the following)

( ) continuation
( ) continuation-in-part
( ) divisional

is being filed in the parent application, from which this application claims priority under 35 U.S.C. § 120.

2.9 Incorporation by Reference

() the entire disclosure of the prior application, from which a copy of the oath or declaration is supplied, is considered to be part of the disclosure of the accompanying application and is hereby incorporated by reference therein.

3. Papers Enclosed Which are Required for Filing Date Under 37 CFR 1.53(b) (Regular) or 37 CFR 1.153 (Design) Application

(X) 15 Pages of specification
(X) 8 Pages of claims
(X) 1 Pages of Abstract
(X) 4 Sheets of drawing
(X) formal
() informal

4. Additional papers enclosed

() Amendment to claims:

( ) **Cancel** in this application claims_____ before calculating the filing fee. (At least one original independent claim must be retained for filing purposes).

( ) **Add** the claims shown in the attached amendment. (Claims added have been numbered consecutively following the highest numbered original claims).

( ) Preliminary Amendment
( ) Information Disclosure Statement (37 C.F.R. 1.98)
( ) Form PTO-1449
( ) Citations
( ) Declaration of Biological Deposit
( ) Special Comments
( ) Other

5. Declaration or oath (including power of attorney)

(X) ENCLOSED.
(X) Newly executed (original or copy)
() Copy from prior application No. 0 /________ (37 CFR 1.63(d)- continuation/divisional)

( ) DELETION OF INVENTOR(S) - signed statement attached deleting inventor(s) named in the above-noted prior application (37 CFR 1.63(d) and 1.33(b))

Declaration or Oath executed by: (check **all** applicable boxes)
(X) inventor(s).
( ) legal representative of inventor(s). 37 CFR 1.42 or 1.43
( ) joint inventor or person showing a proprietary interest on behalf of inventor who refused to sign or cannot be reached.
( ) this is the petition required by 37 CFR 1.47 and the statement required by 37 CFR 1.47 is also attached. See item 13 below for fee.

() NOT ENCLOSED.
( ) Application is made by a person authorized under 37 CFR 1.41(c) on behalf of all the above named inventor(s). The declaration or oath, along with the surcharge required by 37 CFR 1.16(e) can be filed subsequently.
( ) Showing that the filing is authorized. (Not required unless called into question. 37 CFR 1.41(d)).

6. Inventorship Statement

WARNING: If the named inventors are each not the inventors of all the claims an explanation, including the ownership of the various claims at the time the last claimed invention was made, should be submitted.

The inventorship for all the claims in this application are:
() The same

or

() Not the same. An explanation, including the ownership of the various claims at the time the last

claimed invention was made,
() is submitted
() will be submitted.

7. Language

(X) English
() Non-English
() the attached translation is a verified translation. 37 CFR 1.52(d).

8. Assignment

(X) An assignment of the invention to Cisco Technology, Inc.

(X) is attached. (A separate "ASSIGNMENT COVER LETTER ACCOMPANYING NEW PATENT APPLICATION" is also attached.)

() will follow.

() The prior application is assigned of record to __(copy attached).

9. Certified Copy - Foreign Priority Claim Under 35 U.S.C. 119

Certified copy(ies) of application(s)

{list country, application no(s). and filing date(s)}

from which priority is claimed
() is (are) attached.
() will follow.

NOTE: The foreign application forming the basis for the claim for priority **must** be referred to in the **oath** or **declaration**. 37 CFR 1.55(a) and 1.63.

NOTE: This item is for any foreign priority for which the application being filed directly relates. If any parent U.S. application or International Application form which this application claims benefit under 35 U.S.C. 120 is itself entitled to priority from a prior foreign application then complete item 17 on the ADDED PAGES FOR NEW APPLICATION TRANSMITTAL WHERE BENEFIT OR PRIOR U.S. APPLICATION(S) CLAIMED.

10. Fee Calculation (37 C.F.R. 1.16)

A. (X) Regular Application

CLAIMS AS FILED				
	Number Filed	Number Extra	Rate	Basic Fee $690.00

Total Claims (37 CFR 1.16(c))	24 - 20 =	4	x $ 18.00	$72
Independent Claims (37 CFR 1.16(b))	12 - 3 =	9	x $ 78.00	$702
Multiple dependent claim(s), if any (37 CFR 1.16(d))	0	0	+ $ 260.00	$ 0.00

( ) Amendment canceling extra claims enclosed.
( ) Amendment deleting multiple dependencies enclosed.
( ) Fee for extra claims is not being paid at this time.

NOTE: If the fees for extra claims are not paid on filing they must be paid or the claims canceled by amendment, prior to the expiration of the time period set for response by the Patent and Trademark Office in any notice of fee deficiency. 37 CFR 1.16(d).

Filing Fee Calculation $ 1,464.00

B. () Design Application
($330.00 - 37 CFR 1.16(f))

Filing Fee Calculation $

11. Small Entity Statement(s)

() Verified Statement(s) that this is a filing by a small entity under 37 CFR 1.9 and 1.27 is(are) attached.

Filing Fee Calculation (50% of **A** or **B** above) $

NOTE: Any excess of the full fee paid will be refunded if a verified statement and a refund request are filed within **2 months** of the date of timely payment of a full fee. 37 CFR 1.28(a).

12. Request for International-Type Search (37 C.F.R. 1.104(d))

( ) Please prepare an international-type search report for this application at the time when national examination on the merits takes place.

13. Fee Payment Being Made At This Time

( ) NOT ENCLOSED.
( ) No filing fee is to be paid at this time. (This and the surcharge required by 37 CFR 1.16(e) can be paid subsequently.)

(X) ENCLOSED

(X) Filing fee $ 1,464.00

(X) Recording assignment
($40.00; 37 CFR 1.21(h)(1)) $ 40.00

( ) petition fee for filing by other than all the inventors or person on behalf of the inventor where inventor refused to sign or cannot be reached. ($130.00; 37 CFR 1.47 & 1.17(h))
$_____

( ) for processing an application with a specification is a non-English language. ($130.00 37 CFR 1.52(d) and 1.17(k))
$_____

( ) processing and retention fee. ($130.00; 37 CFR 1.53(d) and 1.21(l))
$_____

( ) Fee for international-type search report. ($40.00; 37 CFR 1.21(e))
$_____

Total fees enclosed $ 1,504.00

14. Method of Payment of Fees

(X) Check in the amount of $ 1,504.00

() Charge Account No. 50-0902 in the amount of $ A duplicate of this transmittal is attached.

15. Authorization to Charge Additional Fees

WARNING: If no fees are to be paid on filing the following items should **not** be completed.

WARNING: Accurately count claims, especially multiple dependent claims, to avoid unexpected high charges, if extra claim charges are authorized.

(X) The Commissioner is hereby authorized to charge the following additional fees by this paper and during the entire pendency of this application to Account No. 50-0902, **identifying our Attorney Docket No.** 72255/02663.

(X) 37 CFR 1.16(a), (f), or (g) (filing fees)
(X) 37 CFR 1.16(b), (c) and (d) (presentation of extra claims)
(X) 37 CFR 1.17 (application processing fees)
( ) 37 CFR 1.16(e) (surcharge for filing the basic filing fee and/or declaration on a date later than the filing date of the application)
( ) 37 CFR 1.17(a)(1)-(5) (extension fees pursuant to 37 CFR 1.136(a))

( ) 37 CFR 1.18 (issue fee at or before mailing Notice of Allowance, pursuant to 37 CFR 1.311(b))

16. Instruction As To Overpayment

() Credit Account No. 50-0902, **identifying our Attorney Docket No. ______.**
(X) Refund

17. Incorporation by reference of added pages

(X) The following pages are incorporated by reference:

(X) "Assignment Cover Letter Accompanying New Application"; number of pages added 3

() Added Pages For Papers Referred To In Item 4 Above; number of pages added

( ) Plus added pages deleting names of inventor(s) named in prior application(s) who is/are no longer inventor(s) of the subject matter claimed in this application; number of pages added_ ______.

(X) no further pages form a part of this Transmittal. The transmittal ends with this page.

Date: September 28, 2000

Michael Jaffe

Michael A. Jaffe
Registration No. 36,326
ARTER & HADDEN LLP
1100 Huntington Building
925 Euclid Avenue
Cleveland, Ohio 44115-1475
Phone: (216) 696-3394
Fax: (216) 696-2645
Customer No. 23380

DOCKET NO. 72255/02663

HARDWARE-BASED ENCRYPTION/DECRYPTION EMPLOYING DUAL PORTED KEY STORAGE

Field of Invention

The present invention generally relates to a system for encryption and decryption of data, and more particularly to a hardware-based encryption and decryption system employing dual ported memory for key storage.

Background of the Invention

In a typical wireless LAN configuration, A common model for network processing consists of a multi-level approach. This is common in many Ethernet LAN protocols such as IEEE 802.3. The model typically includes 3 major levels, namely a) Top :Logical-Link control; b) Middle: Medium Access Control (MAC); and c) Bottom: Physical interface (PHY).

A wireless LAN configuration compliant to IEEE 802.11 is similar to its wired counterpart and has the same three levels. Traditionally, the top Logical-Link control tasks are handled by software running on a HOST processor. The middle level is the responsibility of a MAC processor, which handles all frame traffic between the HOST and the PHY level.

In a typical wireless local area network (WLAN) configuration, a portable or mobile device (e.g., a laptop personal computer) normally includes a HOST processor and a PCI card or PCMCIA card. On this card resides a Medium Access Control (MAC) processing system, a PHY (physical layer) processing device (e.g., a digital signal processor), and a main memory. The MAC processing system includes a MAC processor (e.g., an embedded processor), which is a multi-functional processor engine responsible for a variety of different processing tasks associated with the wireless communications. The PHY processing device performs such functions as encoding/decoding waveforms. Data transferred between the PHY processing device and the MAC processing system (i.e., the PHY data stream) may be encrypted using an encryption algorithm, such as RC4. Consequently, encrypted data received by the MAC processing system from the PHY processing device is initially stored to the main memory as encrypted data. At a later time, the MAC processor reads the stored encrypted data from main memory and decrypts the data. The decrypted data is then written to the main memory for subsequent processing by the HOST processor.

Similarly, in the case of a data transmission from the MAC processor to the PHY data processing device, the data originates from the HOST processor that writes the data as plaintext to the main memory. The MAC processor will at a later time read the data from the main memory and encrypt it, using the same RC4 algorithm. Then the encrypted data is transmitted to the PHY processing device.

Encryption algorithm RC4 (developed by RSA Data Security, Inc.) is used to encrypt data in accordance with Wired Equivalent Privacy for the IEEE wireless communications standard 802.11. RC4 is a variable key-size stream cipher with byte-oriented operations. The algorithm is based on the use of a random permutation. Analysis shows that the period of the cipher is overwhelmingly likely to be greater than 10^{100}. Eight to sixteen machine operations are required per output byte, and the cipher can be expected to run very quickly in software. RC4 is commonly used for file encryption and for secure communications, as in the encryption of traffic to and from secure web sites using the secured socket layer (SSL) protocol.

In the prior art, both software and hardware approaches have been used to implement the private key RC4 algorithm. In the case where all operations are performed in software, a simple C program can be used, such as those illustrated herein. The RC4 algorithm can be divided into three basic phases: phase 1 for lookup and loading of a private key; phase 2 having two parts, namely, phase 2a for filling an S-box table linearly and phase 2b for initializing the S-box table with the private key; and phase 3 for the encrypting/decrypting operation (including determination of an X byte and an XOR operation).

It should be understood that Phase 2 of the algorithm includes the two steps of: (phase 2a) filling an S-box table (256 x 8) linearly: S_o=0, S_1=1,...,S_{255}=255; and then (phase 2b) initializing the S-box table by scrambling the table with the private key, repeating the key as necessary to address all 256 locations in the array. For example, if a 16 byte key is used, the sequence would be: KEY_o, KEY_1, ...,KEY_{15}, KEY_o, KEY_1, ...,$KEY_{15,}$ repeating this sequence a total of 16 times to complete the scrambling. It should be appreciated that the term "key" refers to a plurality of "key values." In accordance with a preferred embodiment, each key value is a byte, and a key is comprised of 16 key values (i.e., 16 bytes). 16 bytes are required for 128-bit encryption. The key is a private key known only to the transmitter and receiver(s) of the encrypted data.

As indicated above, in Phase 2b of the RC4 algorithm the S-box table is initialized with the private key. In this regard, index *j* is set to zero, then:

```
For $i = 0$ to 255:
    $j = (j + S_i + KEY_i) \bmod 256$
    swap $S_i$ and $S_j$
```

In the third phase (phase 3), two counters, i and j, are initialized to zero to index through the 256x8 S-box in a pseudorandom fashion. Thereafter, random bytes X are generated as follows:

```
$i = (i+1) \bmod 256$
$j = (j+S_i) \bmod 256$
Swap $S_i$ and $S_j$
$t = (S_i + S_j) \bmod 256$
$X = S_t$
```

The foregoing code sequence is performed for every byte to be encrypted/decrypted. Being a symmetric cryptosystem, the same algorithm is used to decrypt or encrypt data depending on how the XOR data is used. In this regard, the byte X is XORed with plaintext to produce ciphertext or XORed with ciphertext to produce plaintext.

Although the software implementation of the foregoing encryption algorithm appears simple in high level code, the software approach is slow. The software approach is too slow to meet the tight turn around time requirements of IEEE 802.11.

In view of the deficiencies of the software implementation, hardware modifications to the MAC processor have been used to accelerate operation of the algorithm. Referring now to Fig. 2, there is shown a MAC processor 10 according to the prior art. MAC processor 10 is generally comprised of a CPU 20, a key register 30, data path hardware engine 40, microcode controller system 50 (which includes a microcode controller and RAM), and S-Box RAM 100.

CPU 20 is the main processing device of MAC processor 10, and provides signals for controlling operation of key register 30 and microcode controller system 50. Keys are commonly stored in "off-chip" RAM since they are large (often 128 bits or more), and there are many keys that might be used to decrypt and encrypt data depending on the source and destination of the packet address. Therefore, in the prior art a small on-chip key register 30 is used to hold the current key bytes being used. Key register 30 is loaded by using register decodes under the direction of software. Therefore, once the proper key is found, software can load the key and start phase 2 of the encryption algorithm discussed above. Data path hardware engine 40 provides an 8-bit wide data path for performing data manipulation for the RC4 algorithm. Data path hardware engine 40 includes elements such as registers, adders, multiplexers, etc., used to read key register 30 and read/write the S-box table RAM 100. Microcode controller system 50 is used to control the data path to execute the operations needed to execute the RC4 algorithm. S-box table RAM 100 is an "on-chip" RAM (i.e., RAM located on the MAC processor chip) which stores the S-box table. The use of an "on-chip" RAM allows for faster initialization and XOR byte generation than obtained with off-chip memory access.

By using the above-mentioned hardware, the prior art takes a total of 1280 (R/W) microcode operations in order to perform the initialization of the S-box table. In addition, the prior art requires that key loading is totally complete before starting the phase 2b initialization operation. However, phase 2a can be executed at any time since the linear fill of the S-box table does not depend on the key. It should be understood that phase 2a must complete before launching phase 2b.

The present invention provides enhancements to the hardware arrangement in order to further improve the speed of the encryption/decryption operations.

Summary of the Invention

According to a first aspect of the present invention there is provided a method for executing an algorithm for decrypting data, comprising: loading into a memory in consecutive order a plurality of keys respectively associated with a plurality of data frames including encrypted data, wherein each said key is comprised of a plurality of key values; and executing a decryption algorithm simultaneous with loading of key values into the memory, wherein said decryption operation uses key values loaded into memory to decrypt said plurality of data frames.

According to another aspect of the present invention there is provided a method for executing an algorithm for encrypting data, comprising: loading into a memory in consecutive order a plurality of keys respectively associated with a plurality of data frames including unencrypted data, wherein each said key is comprised of a plurality of key values; and executing an encryption algorithm simultaneous with loading of key values into the memory, wherein said decryption operation uses key values loaded into memory to decrypt said plurality of data frames.

According to another aspect of the present invention there is provided a method for executing an algorithm for decrypting data, comprising: loading into a memory in consecutive order a plurality of keys respectively associated with a plurality of data frames including encrypted data, wherein each said key is comprised of a plurality of key values; initializing a table for decryption with at least one of the plurality of key values associated with a first data frame of the plurality of data frames, wherein said initializing step occurs prior to loading of all of the plurality of key values associated with the first data frame, and simultaneous with loading of key values; and executing an algorithm to decrypt the first data frame using the initialized table, said execution occurring simultaneous with loading of key values associated with subsequent data frames of the plurality of data frames.

According to another aspect of the present invention there is provided a method for executing an algorithm for decrypting data, comprising: loading into a memory in consecutive order a plurality of keys respectively associated with a plurality of data frames including encrypted data, wherein each said key is comprised of a plurality of key values; initializing a table for decryption with at least one of the plurality of key values associated with a first data frame of the plurality of data frames, wherein said initializing step occurs prior to loading of all of the plurality of key values associated with the first data frame, and simultaneous with loading of key values; and executing an algorithm to decrypt the first data frame using the initialized table, said execution occurring simultaneous with loading of key values associated with subsequent data frames of the plurality of data frames.

According to another aspect of the present invention there is provided a method for executing an algorithm for encrypting data, comprising: loading into a memory in consecutive order a plurality of keys respectively associated with a plurality of data frames including unencrypted data, wherein each said key is comprised of a plurality of key values; initializing a table for encryption with at least one of the plurality of key values associated with a first data frame of the plurality of data frames, wherein said initializing step occurs prior to loading of all of the plurality of key values associated with the first data frame, and simultaneous with loading of key values; and executing an algorithm to encrypt the first data frame using the initialized table, said execution occurring simultaneous with loading of key values associated with subsequent data frames of the plurality of data frames.

According to another aspect of the present invention there is provided a system for decrypting data, comprising: means for storing in consecutive order a plurality of keys respectively associated with a plurality of data frames including encrypted data, wherein each said key is comprised of a plurality of key values; and means for executing a decryption algorithm simultaneous with storing of key values into the means for storing, wherein said decryption operation uses key values loaded into the means for storing to decrypt said plurality of data frames.

According to another aspect of the present invention there is provided a system for encrypting data, comprising: means for storing in consecutive order a plurality of keys respectively associated with a plurality of data frames including unencrypted data, wherein each said key is comprised of a plurality of key values; and means for executing an encryption algorithm simultaneous with storing of key values into the means for storing, wherein said decryption operation uses key values loaded into the means for storing to decrypt said plurality of data frames.

According to another aspect of the present invention there is provided a system for executing an algorithm for decrypting data, comprising: means for storing in consecutive order a plurality of keys respectively associated with a plurality of data frames including encrypted data, wherein each said key is comprised of a plurality of key values; means for initializing a table for decryption with at least one of the plurality of key values associated with a first data frame of the plurality of data frames, wherein said means for initializing commences initialization of the table prior to loading of all of the plurality of key values associated with the first data frame, and simultaneous with loading of key values; and means for executing an algorithm to decrypt the first data frame using the initialized table, wherein said means for executing decrypts the first data frame simultaneous with storing of key values associated with subsequent data frames of the plurality of data frames.

According to another aspect of the present invention there is provided a system for executing an algorithm for encrypting data, comprising: means for storing in consecutive order a plurality of keys respectively associated with a plurality of data frames including unencrypted data, wherein each said key is comprised of a plurality of key values; means for initializing a table for encryption with at least one of the plurality of key values associated with a first data frame of the plurality of data frames, wherein said initialization occurs prior to loading of all of the plurality of key values associated with the first data frame, and simultaneous with storing of key values; and means for executing an algorithm to encrypt the first data frame using the initialized table, wherein said means for execution executes the algorithm simultaneously with storing of key values associated with subsequent data frames of the plurality of data frames.

According to another aspect of the present invention there is provided a system for decrypting data, comprising: a dual port memory for storing in consecutive order a plurality of keys respectively associated with a plurality of data frames including encrypted data, wherein each said key is comprised of a plurality of key values; and a controller for executing a decryption algorithm simultaneous with storing of key values into the dual port memory, wherein said decryption operation uses key values loaded into the dual port memory to decrypt said plurality of data frames.

According to another aspect of the present invention there is provided a system for encrypting data, comprising: a dual port memory for storing in consecutive order a plurality of keys respectively associated with a plurality of data frames including encrypted data, wherein each said key is comprised of a plurality of key values; and a controller for executing an encryption algorithm simultaneous with storing of key values into the dual port memory, wherein said encryption operation uses key values loaded into the dual port memory to encrypt said plurality of data frames.

According to another aspect the present invention there is provided a system for executing an algorithm for decrypting data, comprising: a dual port memory for storing in consecutive order a plurality of keys respectively associated with a plurality of data frames including encrypted data, wherein each said key is comprised of a plurality of key values; and a controller for: (a) initializing a table for decryption with at least one of the plurality of key values associated with a first data frame of the plurality of data frames, wherein said controller commences initialization of the table prior to loading of all of the plurality of key values associated with the first data frame, and simultaneous with loading of key values, and (b) executing an algorithm to decrypt the first data frame using the initialized table, wherein said controller decrypts the first data frame simultaneous with storing of key values associated with subsequent data frames of the plurality of data frames.

An advantage of the present invention is the provision of a system for encryption/decryption that provides faster lookup of an encryption/decryption private key.

Another advantage of the present invention is the provision of a system for encryption/decryption that provides faster loading of an encryption/decryption private key.

Still another advantage of the present invention is the provision of a system for encryption/decryption that provides faster and more efficient execution of an encryption/decryption algorithm.

Yet another advantage of the present invention is the provision of a system for encryption/decryption that provides greater ease of operation.

Still other advantages of the invention will become apparent to those skilled in the art upon a reading and understanding of the following detailed description, accompanying drawings and appended claims.

Brief Description of the Drawings

The invention may take physical form in certain parts and arrangements of parts, a preferred embodiment and method of which will be described in detail in this specification and illustrated in the accompanying drawings which form a part hereof, and wherein:

Fig. 1 illustrates a typical PHY data stream;

Fig. 2 is a block diagram of a typical prior art hardware configuration for performing encryption/decryption;

Fig. 3 is a block diagram of a hardware configuration for performing encryption/decryption according to a preferred embodiment of the present invention; and

Fig. 4 shows timelines illustrating improvements in processing speed over the prior art.

Detailed Description of the Preferred Embodiment

It should be understood that while a preferred embodiment of the present invention is described in connection with the RC4 encryption/decryption algorithm, the present invention is also suitably applicable to provide improved processing speed and efficiency for other encryption/decryption algorithms that use keys. Moreover, it is contemplated that the present invention has applications outside of data encryption/decryption, as a means for accelerating data processing operations.

Referring now to the drawings wherein the showings are for the purposes of illustrating a preferred embodiment of the invention only and not for purposes of limiting same, Fig. 1 illustrates a typical PHY data stream. The PHY data stream include data frames having two main components, namely, a header portion, and a data portion. The header portion includes control information such as source, destination address; data rate, etc. The MAC processor processes the header information so it knows how to deal with the data portion. Furthermore, some information in the header portion may direct the MAC processor to perform actions such as generating a response message to the sender or steering the data to another station. The header portion is not typically encrypted. The data portion typically includes data that is encrypted, and is intended for processing by the HOST processor. This data may be used by the HOST processor to communicate with other HOST processors across a network using some upper level protocol, such as sending an FTP from one station to another station. In the case of a system complying with IEEE standard 802.11, the HOST processor does not use the header information.

The data processing steps for decryption using prior art systems will now be described in detail with reference to Figs. 1 and 2. As can be seen in Fig. 1, each frame of data is comprised of a header portion and a data portion. A short time gap (i.e., interframe spacing) is provided between each frame. After the header portion of the first data frame (frame #1) has been loaded, phase 1 of the RC4 algorithm commences. The address portion of the header is used to begin the key lookup. After the proper key is found, it is loaded into the key register. While this is happening, encrypted data continues to arrive from the PHY processing device After the key (i.e., the plurality of key values) is entirely loaded into the key register, phase 2 of the RC4 algorithm commences. Phase 2 continues until a period of time after the header for the next data frame (frame #2) has begun to be received. However, most of the time for phase 2 occurs during the interframe spacing. Phase 3 (decrypting/encrypting operation for frame #1) will commence while the header for the next data frame (frame #2) is still being received by the MAC processor. However, as shown, due to the tight turn around time; decryption of all data of frame #1 is not completed before the data portion of frame #2 is received.

This is due to the long period of time needed to complete phases 1 and 2 using the prior art system as shown in Fig. 2. It should be understood that a different key may be used for each received data frame since the MAC processor may be receiving data from different clients, each having a unique private key.

As indicated above, the IEEE wireless communication standard 802.11 requires the use of RC4 to encrypt and decrypt frames that are delivered across a wireless LAN. This creates the potential for performance problems since the inter frame spacing timing (SIFs) between data packets is tight as shown in Fig. 1. The decryption of a frame must complete before a subsequent frame arrives so that the next frame can be decrypted. As described above, the RC4 algorithm uses a multiple step process to prepare an S-box table. This process includes finding and loading a private key, and then scrambling the S-box table to initialize it with the private key. The key loading process cannot begin until the header portion of a frame has been processed. Consequently, there is little time to decrypt the data portion of the frame before the next frame arrives. The problem is more pronounced on short frames since there is little time to "amortize the cost" of loading the key and initializing the S-box table with the key, over the packet size.

Referring now to Fig. 3, there is shown a modified MAC processor 10', according to a preferred embodiment of the present invention. The present invention reduces the amount of time necessary to complete phase 1 (key lookup and load) of the RC4 algorithm. In this regard, the efficiency of Phase 1 is improved by using a key RAM 30' to house the key storage. Key RAM 30' is a dual ported memory which allows a key to be loaded into memory (write port) (phase 1), while simultaneously reading a key (read port) to initialize the S-box table with the keys (phase 2). Consequently, this allows software to start the phase 2 S-box table initialization sequence prior to loading the entire key (i.e., all key values) into memory. Moreover, key RAM 30' is preferably large enough to hold multiple keys (i.e., keys for consecutive data frames). As a result, the key for frame #2 can be loaded into memory, while simultaneously reading out the key for frame #1, without conflict. Thus, overlapping reading and writing is possible which reduces the key load/delay overhead of frame #2, as illustrated in Fig. 1. In this regard, it is not necessary to wait for decryption of frame #1 to be completed before starting to load the key for frame #2 into memory. Key lookup is also improved since it may be possible to predict a forthcoming reception and have the key preloaded as just described. This removes the need to lookup the proper key when the next frame arrives.

It should be appreciated that key RAM 30' may take the form of a dual port synchronous memory (clocked), dual port asynchronous memory (non-clocked), or dual port synchronous burst or non-burst memory.

In accordance with a preferred embodiment of the present invention, phases 2 and 3 of the RC4 algorithm use the same microcoded data path hardware engine 40 as the prior art. Data path hardware engine 40 performs the read/write (R/W) operations to the S-box table RAM 100 to prepare the S-box table.

Fig. 4 provides timelines for a decryption operation that illustrates performance improvements contributed by the present invention. Section I is a timeline associated with the prior art, while Section II is a time line associated with the present invention. It should be appreciated that Section II also illustrates the effects provided by use of level sensitive latches in microcode controller system 50 during microcode fetch and execute, as disclosed in related application serial no. __________________, filed __________________, and assigned to the assignee of the present application.

Six basic steps are performed, namely A - F during two back-to-back decryption phases, as illustrated in Fig. 1. Steps A - F correspond to the tasks noted in the table below:

STEP	**TASK**
A	Lookup and load key for frame #1. May also start phase 2a of frame #1 at this step.

	(phase 1/2a).
B	Initialize S-box table for frame #1. May also do phase 2a of frame #1 at this step. (phase 2a/2b).
C	Lookup and load key for frame #2. May also start phase 2a of frame #2 at this step. (phase 1/2a)
D	Perform decryption for frame #1 (phase 3)
E	Initialize S-box table for frame #2. May also do phase 2a of frame #2 at this step. (phase 2a/2b)
F	Lookup and load key for frame #3. May also start phase 2a of frame #3 at this step. (phase 1)

The three columns in Sections I and II respectively show the resource utilization for loading keys (phase 1), initialization of the S-box table (phase 2) and decryption (phase 3). As can be seen, the prior art takes 160μs to finish with the first decryption (frame #1) and start the next decryption (frame #2). The present invention contributes to completion of the same tasks in 75μs. In this regard, as indicated above, dual-ported key RAM 30' allows a key to be loaded into memory (write port) (phase 1), while simultaneously reading a key (read port) to initialize the S-box table with the keys (phase 2). Thus, the phase 2 S-box table initialization sequence can be started prior to loading the entire key (i.e., all key values) into memory. Moreover, key RAM 30' is preferably large enough to hold multiple keys (i.e., keys for consecutive data frames). As a result, the key for frame #2 can be loaded into memory, while simultaneously reading out the key for frame #1, without conflict.

It can be observed from Fig. 4, that use of dual ported key RAM 30' allows task B to start earlier than in the prior art. In this regard, task A (i.e., loading the key) does not need to be complete before commencing task B. Furthermore, dual ported key RAM '30 allows software to load additional keys for other frames (i.e., tasks C, F, F, F) during execution of task B.

It should be appreciated that in alternative embodiments of the present invention, independent transmission (TX) and reception (RX) engines or a single shared

TX/RX engine may be provided. The encryption/decryption engine, being comprised of memories and datapath, may include a common data path with multiple SBOX and key memories used for TX and RX of forthcoming packets. Another alternative includes the use multiple datapaths and SBOX / key memories for full parallel operations. In this regard it would be possible to perform a full duplex TX encryption and RX decryption at the same time. It should be further appreciated that any number of multiple instances of encryption/decryption engines may be provided on a single MAC processor (e.g., four receive engines, two transmit engines for a total of six different SBOX tables and six different key memories).

It should be understood that the improvements mentioned above in connection with decryption of data are likewise realized for encryption of data. In this regard, the MAC processor benefits from a faster RC4 cycle time and earlier start of phase 2 (i.e., not having to wait for the entire key to be loaded), so that phase 2 of the RC4 algorithm completes sooner than the prior art. Hence, transmission of encrypted data to the PHY processing device can occur at an earlier time. Also, the dual ported key RAM is capable of holding multiple keys allowing preloading of keys for forthcoming transmissions without affecting the current phase 2 operation. Consequently, key lookup and loading are also improved.

The present invention has been described with reference to a preferred embodiment. Obviously, modifications and alterations will occur to others upon a reading and understanding of this specification. It is intended that all such modifications and alterations be included insofar as they come within the scope of the appended claims or the equivalents thereof.

Having thus described the invention, it is now claimed:

1. A method for executing an algorithm for decrypting data, comprising:

loading into a memory in consecutive order a plurality of keys respectively associated with a plurality of data frames including encrypted data, wherein each said key is comprised of a plurality of key values; and

executing a decryption algorithm simultaneous with loading of key values into the memory, wherein said decryption operation uses key values loaded into memory to decrypt said plurality of data frames.

2. A method according to claim 1, said step of executing the decryption algorithm occurring simultaneously with loading of key values associated with subsequent data frames of the plurality of data frames.

3. A method for executing an algorithm for encrypting data, comprising:

loading into a memory in consecutive order a plurality of keys respectively associated with a plurality of data frames including unencrypted data, wherein each said key is comprised of a plurality of key values; and

executing an encryption algorithm simultaneous with loading of key values into the memory, wherein said decryption operation uses key values loaded into memory to decrypt said plurality of data frames.

4. A method according to claim 3, said step of executing the encryption algorithm occurring simultaneously with loading of key values associated with subsequent data frames of the plurality of data frames.

5. A method for executing an algorithm for decrypting data, comprising:

loading into a memory in consecutive order a plurality of keys respectively associated with a plurality of data frames including encrypted data, wherein each said key is comprised of a plurality of key values;

initializing a table for decryption with at least one of the plurality of key values associated with a first data frame of the plurality of data frames, wherein said initializing step occurs prior to loading of all of the plurality of key values associated with the first data frame, and simultaneous with loading of key values; and

executing an algorithm to decrypt the first data frame using the initialized table, said execution occurring simultaneous with loading of key values associated with subsequent data frames of the plurality of data frames.

6. A method according to claim 5, wherein said method further comprises:

initializing said table for decryption with at least one of the plurality of key values associated with a second data frame of the plurality of data frames, wherein said initializing step for the second data frame occurs simultaneous with loading of key values.

7. A method for executing an algorithm for encrypting data, comprising:

loading into a memory in consecutive order a plurality of keys respectively associated with a plurality of data frames including unencrypted data, wherein each said key is comprised of a plurality of key values;

initializing a table for encryption with at least one of the plurality of key values associated with a first data frame of the plurality of data frames, wherein said initializing step occurs prior to loading of all of the plurality of key values associated with the first data frame, and simultaneous with loading of key values; and

executing an algorithm to encrypt the first data frame using the initialized table, said execution occurring simultaneous with loading of key values associated with subsequent data frames of the plurality of data frames.

8. A method according to claim 7, wherein said method further comprises:

initializing said table for encryption with at least one of the plurality of key values associated with a second data frame of the plurality of data frames, wherein said initializing step for the second data frame occurs simultaneous with loading of key values.

9. A system for decrypting data, comprising:

means for storing in consecutive order a plurality of keys respectively associated with a plurality of data frames including encrypted data, wherein each said key is comprised of a plurality of key values; and

means for executing a decryption algorithm simultaneous with storing of key values into the means for storing, wherein said decryption operation uses key values loaded into the means for storing to decrypt said plurality of data frames.

10. A system according to claim 9, wherein said means for executing the decryption algorithm decrypts a data frame simultaneously with storing of key values associated with subsequent data frames of the plurality of data frames, in said means for storing.

11. A system for encrypting data, comprising:

means for storing in consecutive order a plurality of keys respectively associated with a plurality of data frames including unencrypted data, wherein each said key is comprised of a plurality of key values; and

means for executing an encryption algorithm simultaneous with storing of key values into the means for storing, wherein said decryption operation uses key values loaded into the means for storing to decrypt said plurality of data frames.

12. A system according to claim 11, wherein said means for executing the decryption algorithm encrypts a data frame simultaneously with storing of key values associated with subsequent data frames of the plurality of data frames, in said means for storing.

13. A system for executing an algorithm for decrypting data, comprising:

means for storing in consecutive order a plurality of keys respectively associated with a plurality of data frames including encrypted data, wherein each said key is comprised of a plurality of key values;

means for initializing a table for decryption with at least one of the plurality of key values associated with a first data frame of the plurality of data frames, wherein said means for initializing commences initialization of the table prior to loading of all of the plurality of key values associated with the first data frame, and simultaneous with loading of key values; and

means for executing an algorithm to decrypt the first data frame using the initialized table, wherein said means for executing decrypts the first data frame simultaneous with storing of key values associated with subsequent data frames of the plurality of data frames.

14. A system according to claim 13, wherein said system further comprises:

means for initializing said table for decryption with at least one of the plurality of key values associated with a second data frame of the plurality of data frames, wherein said initialization of the second data frame occurs simultaneously with storing of key values.

15. A system for executing an algorithm for encrypting data, comprising:

means for storing in consecutive order a plurality of keys respectively associated with a plurality of data frames including unencrypted data, wherein each said key is comprised of a plurality of key values;

means for initializing a table for encryption with at least one of the plurality of key values associated with a first data frame of the plurality of data frames, wherein said initialization occurs prior to loading of all of the plurality of key values associated with the first data frame, and simultaneous with storing of key values; and

means for executing an algorithm to encrypt the first data frame using the initialized table, wherein said means for execution executes the algorithm simultaneously with storing of key values associated with subsequent data frames of the plurality of data frames.

16. A system according to claim 15, wherein said system further comprises:

means for initializing said table for encryption with at least one of the plurality of key values associated with a second data frame of the plurality of data frames, wherein said initialization for the second data frame occurs simultaneously with loading of key values.

17. A system for decrypting data, comprising:

a dual port memory for storing in consecutive order a plurality of keys respectively associated with a plurality of data frames including encrypted data, wherein each said key is comprised of a plurality of key values; and

a controller for executing a decryption algorithm simultaneous with storing of key values into the dual port memory, wherein said decryption operation uses key values loaded into the dual port memory to decrypt said plurality of data frames.

18. A system according to claim 17, wherein said controller decrypts a data frame simultaneously with storing of key values associated with subsequent data frames of the plurality of data frames, in said dual port memory.

19. A system for encrypting data, comprising:

a dual port memory for storing in consecutive order a plurality of keys respectively associated with a plurality of data frames including encrypted data, wherein each said key is comprised of a plurality of key values; and

a controller for executing an encryption algorithm simultaneous with storing of key values into the dual port memory, wherein said encryption operation uses key values loaded into the dual port memory to encrypt said plurality of data frames.

20. A system according to claim 19, wherein said controller encrypts a data frame simultaneously with storing of key values associated with subsequent data frames of the plurality of data frames, in said dual port memory.

21. A system for executing an algorithm for decrypting data, comprising:

a dual port memory for storing in consecutive order a plurality of keys respectively associated with a plurality of data frames including encrypted data, wherein each said key is comprised of a plurality of key values; and

a controller for:

(a) initializing a table for decryption with at least one of the plurality of key values associated with a first data frame of the plurality of data frames, wherein said controller commences initialization of the table prior to loading of all of the plurality of key values associated with the first data frame, and simultaneous with loading of key values, and

(b) executing an algorithm to decrypt the first data frame using the initialized table, wherein said controller decrypts the first data frame simultaneous with storing of key values associated with subsequent data frames of the plurality of data frames.

22. A system according to claim 21, wherein said controller initializes said table for decryption with at least one of the plurality of key values associated with a second data frame of the plurality of data frames, wherein said initialization of the second data frame occurs simultaneously with storing of key values.

23. A system for executing an algorithm for encrypting data, comprising:

a dual port memory for storing in consecutive order a plurality of keys respectively associated with a plurality of data frames including unencrypted data, wherein each said key is comprised of a plurality of key values; and

a controller for:

(a) initializing a table for encryption with at least one of the plurality of key values associated with a first data frame of the plurality of data frames, wherein said controller commences initialization of the table prior to loading of all of the plurality of key values associated with the first data frame, and simultaneous with loading of key values, and

(b) executing an algorithm to encrypt the first data frame using the initialized table, wherein said controller encrypts the first data frame simultaneous with storing of key values associated with subsequent data frames of the plurality of data frames.

24. A system according to claim 23, wherein said controller initializes said table for encryption with at least one of the plurality of key values associated with a second data frame of the plurality of data frames, wherein said initialization of the second data frame occurs simultaneously with storing of key values.

ABSTRACT

A system for the encryption and decryption of data employing dual ported RAM for key storage to accelerate data processing operations. The on-chip key storage includes a dual-ported memory device which allows keys to be loaded into memory simultaneous with keys being read out of memory. Thus, an encryption or decryption algorithm can proceed while keys are being loaded into memory.

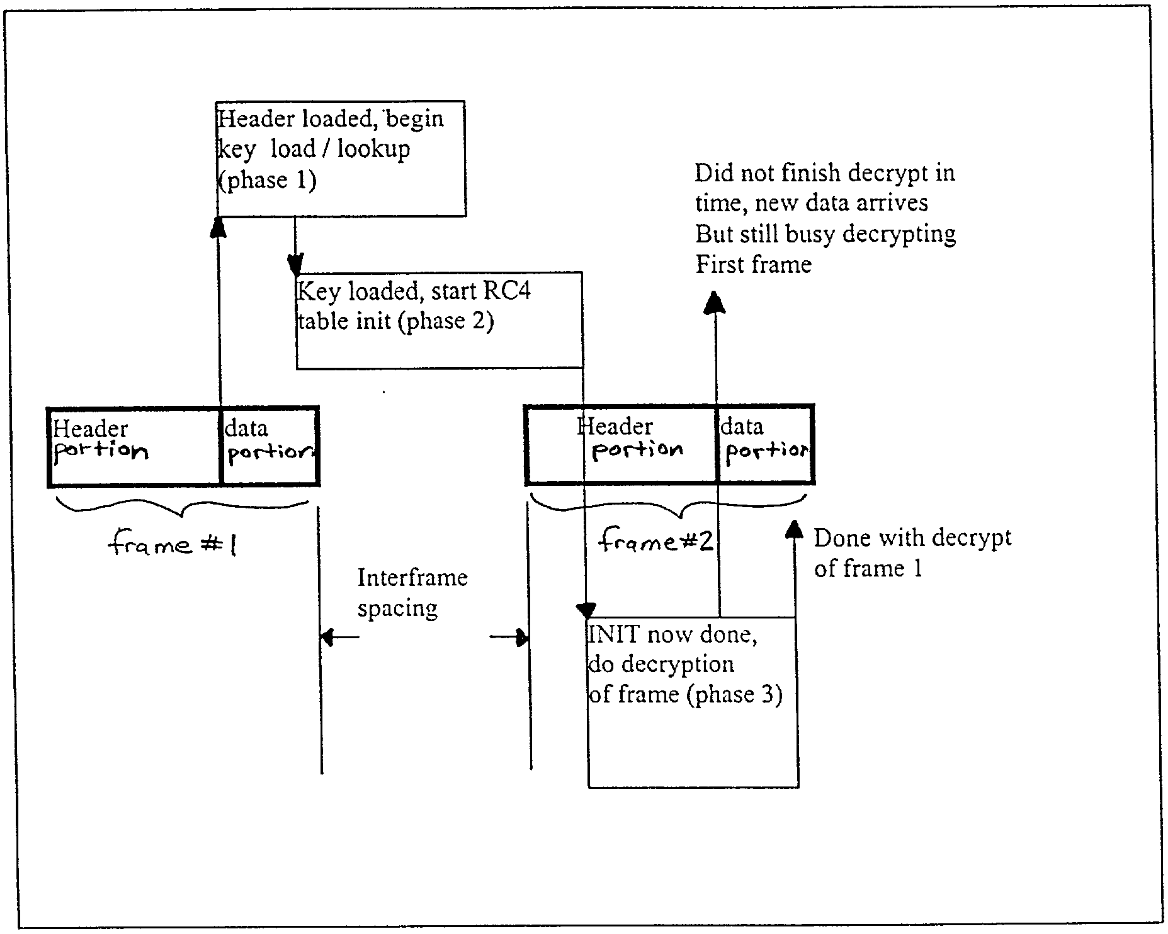

Header loaded, begin key load / lookup (phase 1)
Did not finish decrypt in time, new data arrives But still busy decrypting First frame
Key loaded, start RC4 table init (phase 2)
Header portion
data portion
Header portion
data portion
frame #1
frame #2
Done with decrypt of frame 1
Interframe spacing
INIT now done, do decryption of frame (phase 3)

Fig. 1

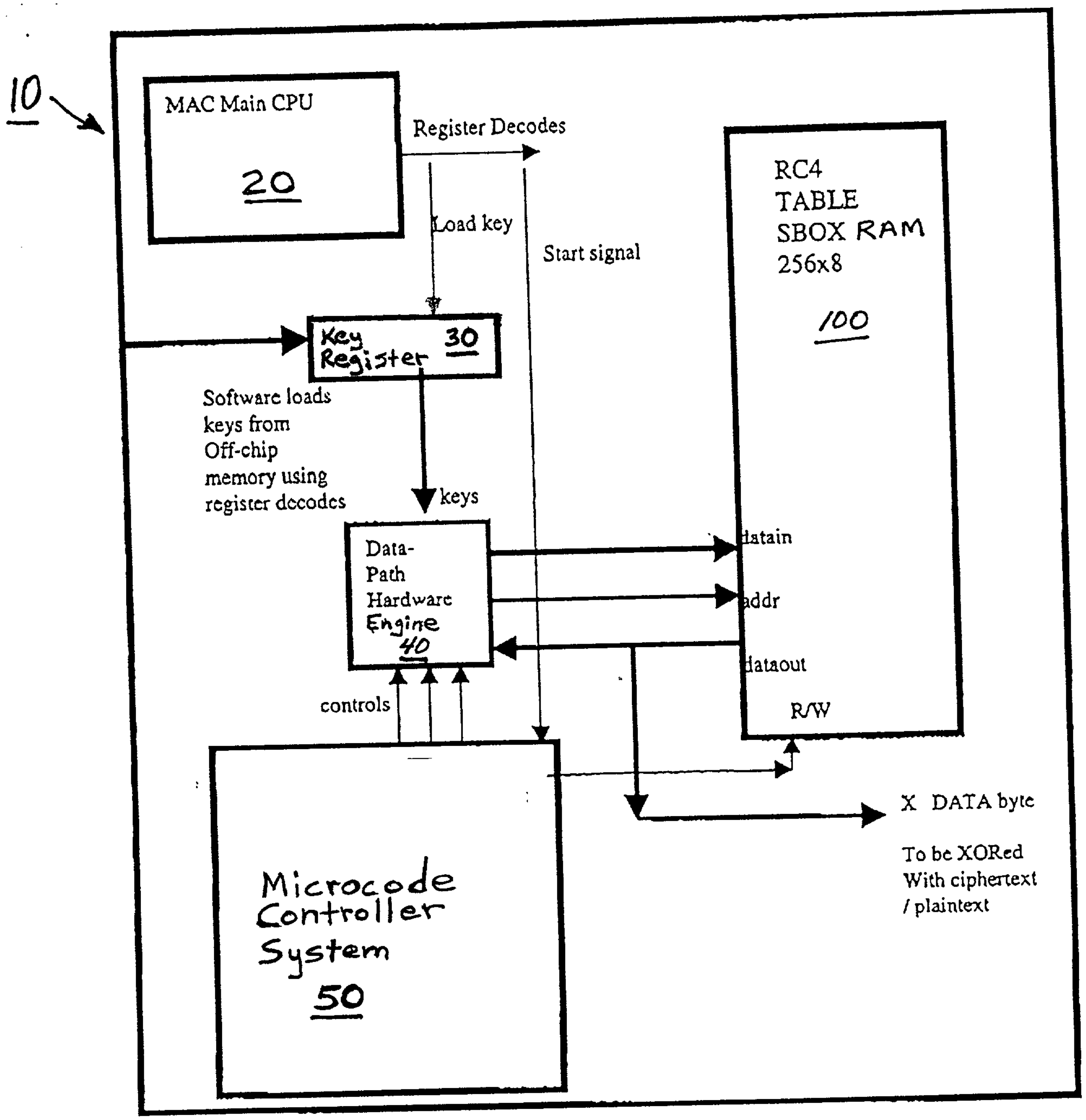
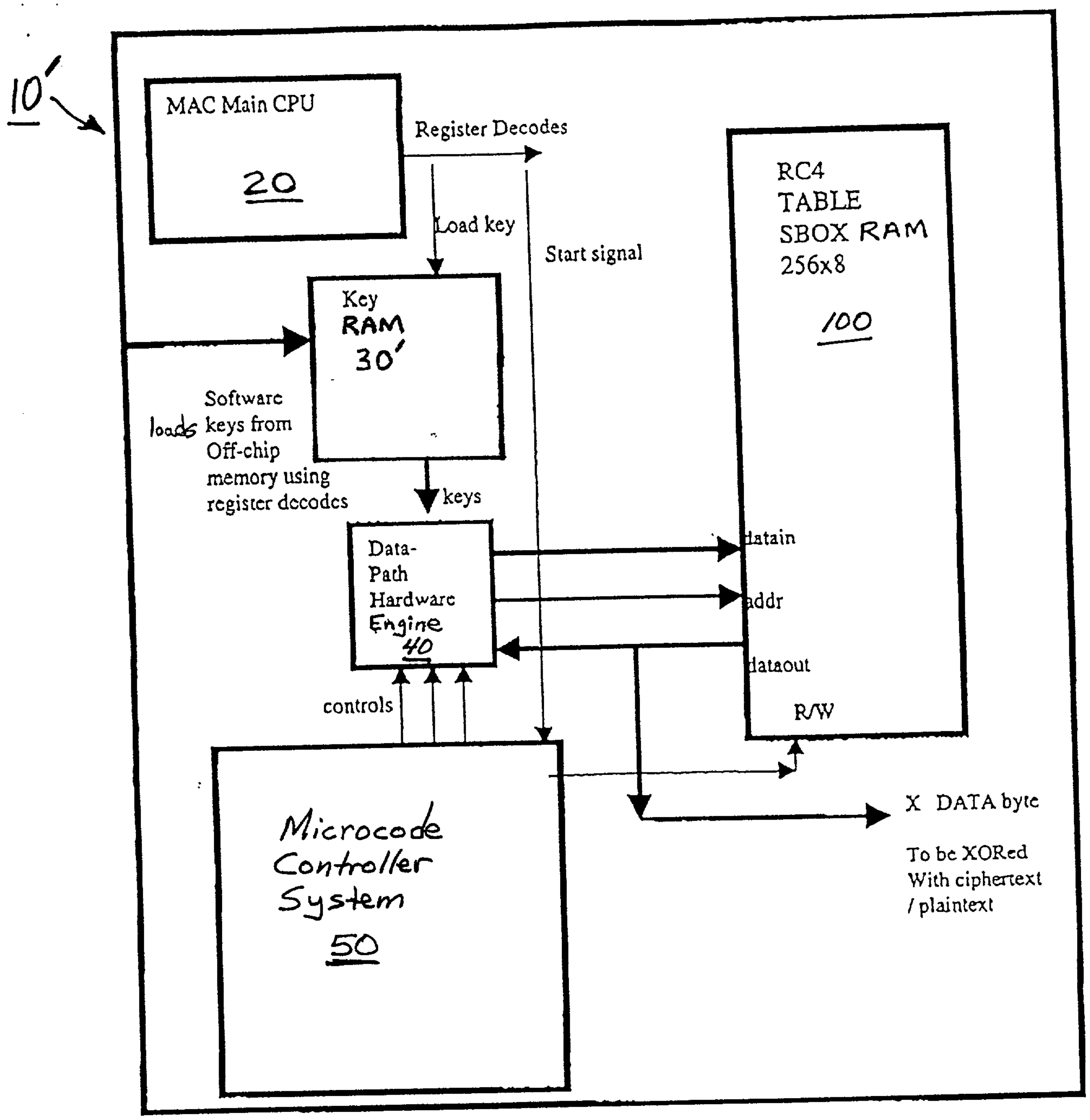


Fig. 2

PRIOR ART



10'
MAC Main CPU
20
Register Decodes
Load key
Start signal
RC4
TABLE
SBOX RAM
256x8
100
Key
RAM
30'
Software
loads keys from
Off-chip
memory using
register decodes
keys
Data-
Path
Hardware
Engine
40
datain
addr
dataout
R/W
controls
Microcode
Controller
System
50
X DATA byte
To be XORed
With ciphertext
/ plaintext

Fig. 3

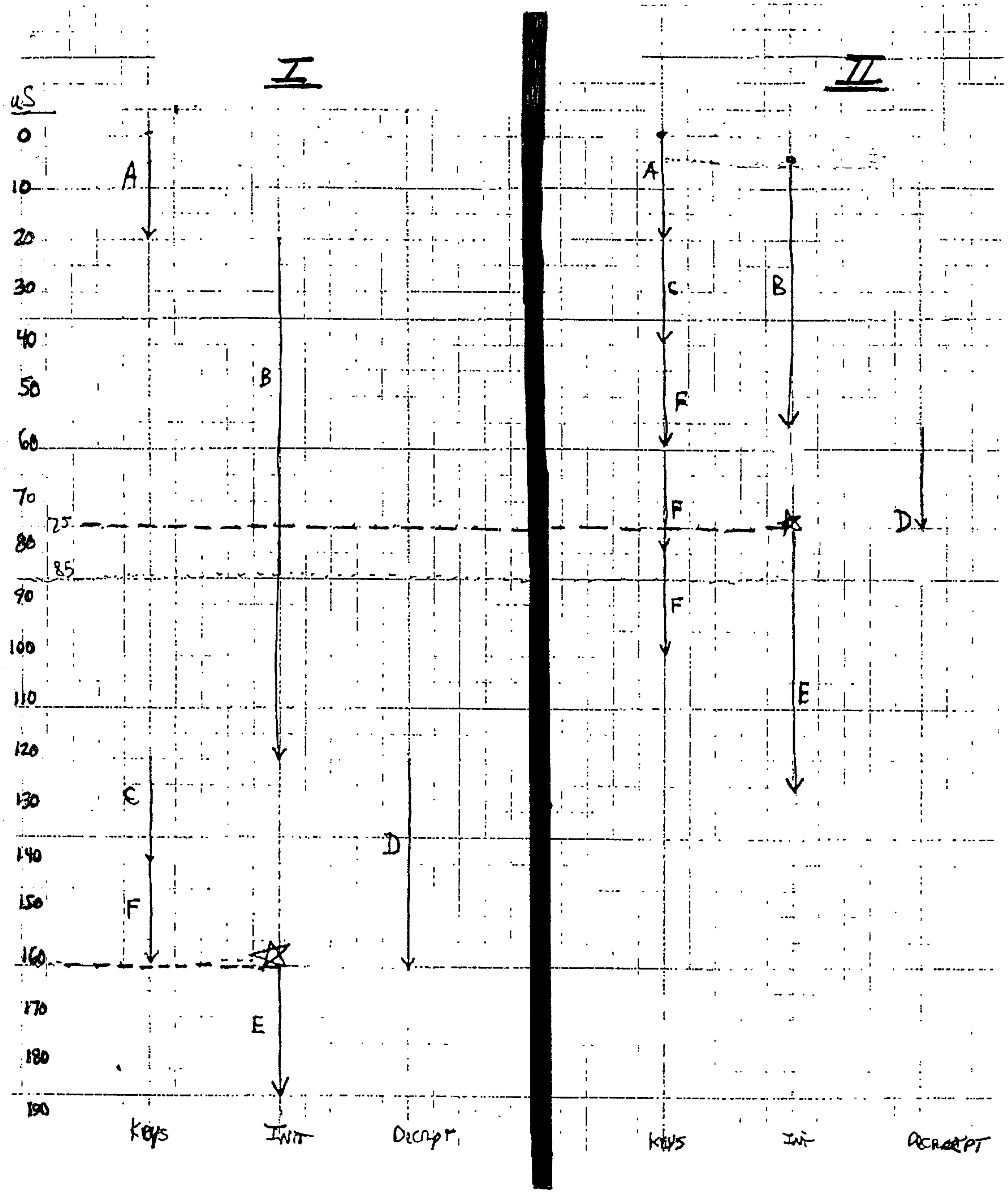

I
II
uS
0
10
20
30
40
50
60
70
75
80
85
90
100
110
120
130
140
150
160
170
180
190
A
B
C
F
E
D
A
C
F
F
F
B
D
E
Keys
Init
Decrypt
Keys
Init
Decrypt

Fig. 4

Docket No. 72255/02663

DECLARATION FOR PATENT APPLICATION

As a below named inventor, I hereby declare that:

My residence, post office address, and citizenship are as stated below next to my name.

I believe that I am the original, first and sole inventor (if only one name is listed below), or an original, first and joint inventor (if plural names are listed below) of the subject matter which is claimed, and for which a patent is sought on the invention entitled:

HARDWARE-BASED ENCRYPTION/DECRYPTION EMPLOYING DUAL PORTED KEY STORAGE

the specification of which is attached hereto, unless the following box is checked:

__ was filed on ____________________, 20__, as United States Application Number or PCT International Application Number ____________________.

I hereby state that I have reviewed and understand the contents of the above-identified specification, including the claims, as amended by any amendment referred to above.

I acknowledge the duty to disclose information which is material to patentability as defined in Title 37, Code of Federal Regulations, §1.56.

I hereby claim foreign priority benefits under Title 35, United States Code, §§119(a) - (d) or §365(b) of any foreign application(s) for patent or inventor's certificate, or §365(a) of any PCT International application which designated at least one country other than the United States, listed below and have also identified below any foreign application for patent or inventor's certificate, or PCT International application having a filing date before that of the application on which priority is claimed:

NONE

I hereby claim the benefit under Title 35, United States Code, §119(e) of any United States provisional application(s) listed below:

NONE

I hereby claim the benefit under Title 35, United States Code, §120 of any United States application(s), or §365(c) of any PCT International application designating the United States, listed below and, insofar as the subject matter of each of the claims of this application is not disclosed in the prior United States or PCT International application in the manner provided by the first paragraph of Title 35, United States Code, §112, I acknowledge the duty to disclose information which is material to patentability as defined in Title 37, Code of Federal Regulations, §1.56 which became available between the filing date of the prior application and the national or PCT International filing date of this application:

NONE

I hereby appoint the following registered attorney(s) and/or agent(s) to prosecute this application and to transact all business in the Patent and Trademark Office connected therewith:

Alan J. Ross, Reg. No. 33,767
John X. Garred, Reg. No. 31,830
Michael A. Jaffe, Reg. No. 36,326
Susan L. Mizer, Reg. No. 38,245
James C. Scott, Reg. No. 35,351
Jay P. Ryan, Reg. No. 37,064

Direct all telephone calls to : Michael A. Jaffe
at telephone number : (216) 696-3394

Direct all correspondence to : Michael A. Jaffe

ARTER & HADDEN LLP
1100 Huntington Building
925 Euclid Avenue
Cleveland, Ohio 44115-1475
Customer No. 23380

I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under 18 U.S.C. 1001 and that such willful false statements may jeopardize the validity of the application or any patent issued thereon.

Full name of sole or first inventor: Kenneth W. Batcher

Inventor's signature : Kenneth W. Batcher

Date : 9/28/00

Residence : Hudson, Ohio
Citizenship : United States of America
Post Office Address : 1769 Mayflower Lane, Hudson, Ohio 44236